

Article ID:1005-3085(2009)05-0781-05

A Continuous Algorithm for Max-bisection*

LI Yu¹, LING Ai-fan^{2,3}

(1- School of Economics and Management, Xinyang Normal University, Xinyang 464000;

2- School of Finance, Jiangxi Finance & Economics University, Nanchang 330013;

3- School of Science, Xi'an Jiaotong University, Xi'an 710049)

Abstract: A continuous algorithm for max-bisection is proposed. We first convert the max-bisection problem to a nonlinear program, then the resulted problem is solved by using the augmented Lagrange penalty function method.

Keywords: combinatorial optimization; max-bisection; penalty function; NCP

Classification: AMS(2000) 90C27 **CLC number:** O221.7 **Document code:** A

1 Introduction

Let $G(V; E)$ be an undirected weighted graph, where $V = \{1, \dots, n\}$ is the set of nodes, E is the set of edges. Let $W = (w_{ij})$ be its weighted adjacency matrix with $w_{ij} > 0$ for $(i, j) \in E$ and $w_{ij} = 0$ for $(i, j) \notin E$. Max-bisection consists in finding a partition of the set V into two sets $S, T = V \setminus S$, such that $|S| = |T| = n/2$ and the total weight of edges with endpoints in different subsets is maximized, i.e.

$$w(S) = \max \sum_{i \in S, j \in T} w_{ij},$$

when $n = |V|$ is even. The problem can be formulated by assigning each node a binary variable $x_i \in \{-1, 1\}$

$$\text{MB: } \max \left\{ f(\mathbf{x}) = \frac{1}{4} \sum_{i,j} w_{ij}(1 - x_i x_j) = \mathbf{x}^T L \mathbf{x} : \mathbf{e}^T \mathbf{x} = 0, \mathbf{x} \in \{-1, 1\}^n \right\},$$

where

$$\mathbf{x} = (x_1, x_2, \dots, x_n)^T, \quad S = \{i | x_i = 1\}, \quad \mathbf{e} = (1, 1, \dots, 1) \in \mathbf{R}^n, \quad L = \frac{1}{4}(\text{Diag}(W\mathbf{e}) - W),$$

$\text{Diag}(\mathbf{a})$ means a diagonal matrix whose diagonal entries are vector \mathbf{a} . The constraint $\mathbf{e}^T \mathbf{x} = 0$ ensures that

$$|S| = |T| = \frac{n}{2}.$$

Frieze & Jerrum^[1] extended Goemans-Williamson's approach^[2] to max-bisection and obtained an 0.651-approximation algorithm. Ye in [4] improved this result to 0.699. Recently, Xu et al.^[3] proposed a feasible direction algorithm without linear search in which max-bisection was relaxed into a quadratic programming problem with nonlinear constraints

$$XMB: \max \{ \mathbf{x}^T L \mathbf{x} : \Phi_F(1 - x_j, 1 + x_j) \leq 0, j = 1, \dots, n; \mathbf{e}^T \mathbf{x} = 0; \|\mathbf{x}\|_2 = 1 \},$$

Received date: 2007-10-16.

Biography: Li Yu (Born in 1963), Male, Ph.D. Research field: combinatorial optimization and its application in financial optimization.

***Foundation item:** The National Natural Science Foundation of China (10671152).

where $\Phi_F(a, b) = \sqrt{a^2 + b^2} - (a + b)$, $a, b \in \mathbf{R}$ is the FB function.

2 Equivalent nonlinear program

A function $\Phi: \mathbf{R}^2 \rightarrow \mathbf{R}$ is called an NCP-function if $\Phi(a, b) = 0 \Leftrightarrow ab = 0, a \geq 0, b \geq 0$. By this definition, we define $\Phi(a, b) = \max(0, ab)^3 + \max(0, -a)^3 + \max(0, -b)^3$. Clearly, $\Phi(a, b)$ is a NCP function. Let $a = 1 - x_i, b = 1 + x_i, i = 1, \dots, n$, then $\Phi(1 - x_i, 1 + x_i) = 0 \Leftrightarrow x_i \in \{-1, 1\}$. Moreover, it is easy to show $\Phi(1 - x_i, 1 + x_i)$ has up to the second order derivation with respect to $x_i, i = 1, \dots, n$. Hence, MB can be rewritten as

$$\text{MB1: } \max\{f(\mathbf{x}) = \mathbf{x}^T L \mathbf{x} : \mathbf{e}^T \mathbf{x} = 0; \Phi(1 - x_i, 1 + x_i) = 0, i = 1, \dots, n\}.$$

Let $\lambda_{\max}(L)$ denote the maximum eigenvalue of matrix L , and $\rho = \frac{1}{4}n\lambda_{\max}(L)$. Consider the following problem

$$\text{MB2: } \max\{F(\mathbf{x}) = \mathbf{x}^T L \mathbf{x} - \rho \mathbf{x}^T \mathbf{e} \mathbf{e}^T \mathbf{x} = \mathbf{x}^T (L - \rho \mathbf{e} \mathbf{e}^T) \mathbf{x} : \Phi(1 - x_i, 1 + x_i) = 0, i = 1, \dots, n\}.$$

Then we have the following results.

Lemma 2.1 Let \mathbf{x}^{**} be an optimization solution of problem MB2. Then \mathbf{x}^{**} is also an optimization solution of problem MB1, also MB2 and MB1 have the same optimization value at \mathbf{x}^{**} .

Lemma 2.1 shows that, instead of solving problem MB1 directly, we can also obtain its optimization solution by solving problem MB2. Let $C = -(L - \rho \mathbf{e} \mathbf{e}^T)$. Then MB2 can be written as

$$\text{MB3: } \min\{h(\mathbf{x}) = \mathbf{x}^T C \mathbf{x} = \mathbf{x}^T (\rho \mathbf{e} \mathbf{e}^T - L) \mathbf{x} : \Phi(1 - x_i, 1 + x_i) = 0, i = 1, \dots, n\}.$$

The augmented Lagrange penalty function method (ALPFM) of problem MB3 has the form

$$\mathcal{P}(\mathbf{x}; \lambda, \sigma) = h(\mathbf{x}) - \lambda^T \Psi(\mathbf{x}) + \frac{\sigma}{2} \|\Psi(\mathbf{x})\|^2,$$

where

$$\Psi(\mathbf{x}) = (\psi(x_1), \dots, \psi(x_n))^T = (\Phi(1 - x_1, 1 + x_1), \dots, \Phi(1 - x_n, 1 + x_n))^T,$$

$\lambda = (\lambda_1, \dots, \lambda_n)^T$ is the Lagrange multiplier vector, and σ is a penalty factor. The function $\mathcal{P}(\mathbf{x}; \lambda, \sigma)$ has up to second order derivative and the gradient of the function $\mathcal{P}(\mathbf{x}; \lambda, \sigma)$ with respect to \mathbf{x} is easily evaluated by

$$\nabla_{\mathbf{x}} \mathcal{P}(\mathbf{x}; \lambda, \sigma) = 2C\mathbf{x} - \nabla \Psi(\mathbf{x})(\lambda - \sigma \Psi(\mathbf{x})),$$

where

$$\nabla \Psi(\mathbf{x}) = \text{Diag}\left(\frac{d\psi(x_1)}{dx_1}, \dots, \frac{d\psi(x_n)}{dx_n}\right).$$

3 Solving problem MB3

Let \mathbf{x}^* and λ^* be an optimal solution of problem MB2 and the associated Lagrangian multiplier vector, respectively. The KKT condition at $(\mathbf{x}^*, \lambda^*)$ of problem MB2 can be presented as

$$\nabla F(\mathbf{x}^*) - \nabla \Psi(\mathbf{x}^*) \lambda^* = 0. \quad (1)$$

Let $\{\lambda^k\}_{k=0}^\infty$ be a given sequence. For each given λ^k , we can minimize $\mathcal{P}(\mathbf{x}; \lambda^k, \sigma)$ for large enough σ . Assume that \mathbf{x}^{k+1} is a minimizer of $\mathcal{P}(\mathbf{x}; \lambda^k, \sigma)$, then the sequence $\{\lambda^k\}_{k=0}^\infty$, for large enough σ , is modified by

$$\lambda^{k+1} = \lambda^k - \sigma \Psi(\mathbf{x}^{k+1}), \quad k = 0, 1, \dots$$

Hence, the minimizer \mathbf{x}^* of MB2 can be obtained by minimizing the function sequence $\mathcal{P}(\mathbf{x}; \lambda^k, \sigma)$. Summarize the statement above, we can described the algorithm for MB3 as follows.

Algorithm CMB:

- 1) Choose randomly an initial point $\mathbf{x}^0 \in (-a, a)^n$, $a \geq 1$, given σ_0, λ^0 , $0 < \epsilon < 1$ and $0 < \rho < 1$, set $k = 0$;
- 2) Start from \mathbf{x}^k , minimize $\mathcal{P}(\mathbf{x}; \lambda^k, \sigma_k)$ to obtain a solution \mathbf{x}^{k+1} . If $\|\Psi(\mathbf{x}^{k+1})\| \leq \epsilon$, set $\mathbf{x}^* = \mathbf{x}^{k+1}$, goto 5);
- 3) If $\|\Psi(\mathbf{x}^{k+1})\| > \rho \|\Psi(\mathbf{x}^k)\|$, set $\sigma_k = 10\sigma_k$ and $\mathbf{x}_k = \mathbf{x}_{k+1}$, goto 2);
- 4) If $\|\Psi(\mathbf{x}^{k+1})\| \leq \rho \|\Psi(\mathbf{x}^k)\|$, set $\lambda^{k+1} = \lambda^k - \sigma_k \Psi(\mathbf{x}^{k+1})$, $\sigma_{k+1} = \sigma_k$, and $k = k + 1$, goto 2);
- 5) Construct a bisection.
 - (a) If $|S| = \frac{n}{2}$, then output S is an optimization bisection partition and stop;
 - (b) If $|S| > \frac{n}{2}$, then for each node $i \in S$, compute

$$\delta(i) = \sum_{j \in S} w_{ij} - \sum_{j \in T} w_{ij},$$

and rearrange the set $S = \{i_1, i_2, \dots, i_{|S|}\}$, where $\delta(i_1) \leq \delta(i_2) \leq \dots \leq \delta(i_{|S|})$.

Let $S = \{i_1, i_2, \dots, i_{\frac{n}{2}}\}$;

- (c) If $|S| < \frac{n}{2}$, then set $\mathbf{x}^* = -\mathbf{x}^*$, goto step 5)(b).

The solution sequence $\{\mathbf{x}^k\}_{k=0}^\infty$ obtained by the algorithm CMB generates a sequence $\{\Psi(\mathbf{x}^k)\}_{k=0}^\infty$ that converges to zero at a rate less than ρ , the condition $\|\Psi(\mathbf{x}^{k+1})\| \leq \epsilon$ can be satisfied in a finite number of iterations (see [3]).

Theorem 3.1 If $\{\lambda^k\}_{k=0}^\infty$ is a bounded sequence and \mathbf{x}^k is a global minimizer of $\mathcal{P}(\mathbf{x}; \lambda^{k-1}, \sigma_k)$, then any accumulation point \mathbf{x}^* of the infinite sequence $\{\mathbf{x}^k\}_{k=0}^\infty$ generated by the algorithm CMB with termination criterion $\|\Psi(\mathbf{x}^k)\| = 0$ is an optimal solution of problem MB3.

Proof By algorithm CMB, the infinite sequence $\{\mathbf{x}^k\}_{k=0}^\infty$ generated by the algorithm CMB satisfies

$$\|\Psi(\mathbf{x}^{k+1})\| \leq \rho \|\Psi(\mathbf{x}^k)\|, \quad 0 < \rho < 1.$$

That is

$$\lim_{k \rightarrow \infty} \|\Psi(\mathbf{x}^k)\| = 0.$$

Hence, any accumulation point of the infinite sequence $\{\mathbf{x}^k\}_{k=0}^\infty$ must be a feasible solution of MB3.

Let $\{\mathbf{x}^{k_j}\}$ be a convergent subsequence of $\{\mathbf{x}^k\}_{k=0}^\infty$ and \mathbf{x}^* its limitation point. Then \mathbf{x}^* is a feasible solution of MB3. Let \mathbf{x}^{**} be a global optimal solution of MB3. Since \mathbf{x}^{k_j} is a global minimizer of function $\mathcal{P}(\mathbf{x}; \lambda^{k_j-1}, \sigma_{k_j})$ and \mathbf{x}^{**} is only a feasible solution of $\mathcal{P}(\mathbf{x}; \lambda^{k_j-1}, \sigma_{k_j})$, we have, for sufficient large k_j ,

$$\begin{aligned} \mathcal{P}(\mathbf{x}^{k_j}; \lambda^{k_j-1}, \sigma_{k_j}) &= h(\mathbf{x}^{k_j}) - (\lambda^{k_j-1})^T \Psi(\mathbf{x}^{k_j}) + \frac{\sigma_{k_j}}{2} \|\Psi(\mathbf{x}^{k_j})\|^2 \\ &\leq \mathcal{P}(\mathbf{x}^{**}; \lambda^{k_j-1}, \sigma_{k_j}) = h(\mathbf{x}^{**}). \end{aligned} \quad (2)$$

Since $\{\lambda^k\}_{k=0}^\infty$ is a bounded sequence, for sufficient k , it follows from (2) that

$$h(\mathbf{x}^{k_j}) + o(1) \leq h(\mathbf{x}^{**}). \quad (3)$$

Let $k_j \rightarrow \infty$ in the left hand side of (3), it follows that

$$h(\mathbf{x}^*) = \lim_{k \rightarrow \infty} h(\mathbf{x}^{k_j}) \leq h(\mathbf{x}^{**}).$$

On the other hand, by the feasibility of \mathbf{x}^* , $h(\mathbf{x}^*) \geq h(\mathbf{x}^{**})$. Therefore $h(\mathbf{x}^*) = h(\mathbf{x}^{**})$ which means \mathbf{x}^* is an optimal solution of MB3.

4 Numerical experiments

For minimizing $\mathcal{P}(\mathbf{x}; \lambda^{k-1}, \sigma_k)$ with fixed λ^{k-1}, σ_k , we use the function *fminunc* in Matlab.

The test problems are some randomly generated graphs. Let $p \in (0, 1)$ be fixed. For each pair (i, j) , $i \neq j$, $c \in (0, 1)$ is random number. If $c \leq p$, then there is an edge between nodes i and j , and set w_{ij} is a random integer number in $[1, 10]$. Otherwise, set $w_{ij} = 0$. We compare our algorithm with the 0.699-approximation algorithm, see Table 1. From the numerical results, one is not hard to see that the proposed continuous algorithm can generate a better solution than the 0.699-approximation algorithm for most of these problems.

Table 1: Comparisons of 0.699-appro. algorithm and CMB

Pro.		SDP	0.699		CMB		
n	p	UB	F_Y	t_Y	F_C^*	t_C	σ
14	0.60	19.2	17	0.31	19	0.68	10^3
16	0.30	10.1	9	0.89	10	1.02	10^3
20	0.60	31.5	25	1.31	31	2.06	10^4
24	0.30	28.1	24	1.25	27	2.12	10^4
30	0.60	76.6	71	1.95	75	2.56	10^4
36	0.60	116.5	109	2.15	114	2.96	10^4
40	0.30	75.9	71	2.35	75	3.12	10^4

References:

- [1] Frieze A, Jerrum M. Improved approximation algorithm for max k -cut and max-bisection[J]. Algorithmica, 1997, 18: 67-81
- [2] Goemans M X, Williamson D P. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming[J]. J Assoc Comput Mach, 1995, 42(6): 1115-1145
- [3] Xu F M, Xu C X, Xue H G. A feasible direction algorithm without line search for solving max-bisection problem[J]. Journal of Computational Mathematics, 2005, 23: 619-634
- [4] Ye Y. A 0.699-approximation algorithm for max-bisection[J]. Mathematical Programming, 2001, 90: 101-111

一种求解最大二等分问题的连续化算法

李 毓¹, 凌爱凡^{2,3}

(1- 信阳师范学院经济与管理学院, 信阳 464000;

2- 江西财经大学金融学院, 南昌 330013; 3- 西安交通大学理学院, 西安 710049)

摘 要: 本文提出了一种求解最大二等分问题的连续化算法。我们首先将二等分问题转化为一个非线性规划; 然后通过增广 Lagrange 罚函数方法来求解这个非线性规划问题。

关键词: 组合最优化; 最大二等分; 罚函数; NCP